

Fault-Tolerant Quantum Walks

S. D. Freedman, Y. Tong, and J. B. Wang*

School of Physics, The University of Western Australia, Perth, Australia

Quantum walks are expected to serve important modelling and algorithmic applications in many areas of science and mathematics. Although quantum walks have been successfully implemented physically in recent times, no major efforts have been made to combat the error associated with these physical implementations in a fault-tolerant manner. In this paper, we propose a systematic method to implement fault-tolerant quantum walks in discrete time on arbitrarily complex graphs, using quantum states encoded with the Steane code and a set of universal fault tolerant matrix operations.

PACS numbers: 03.67.-a, 03.67.Pp, 03.67.Lx

* jingbo.wang@uwa.edu.au

I. INTRODUCTION

Classical random walks have been used in the process of modelling natural systems in fields such as biology [1] and chemistry [2]. They have also been used in applied mathematics, in areas including optimization [3] and solving differential equations [4]. Quantum walks, the quantum analogue to classical random walks, will likely provide a more efficient way to carry out these processes, due to their non-intuitive and markedly different properties, including faster mixing and hitting times and resonance tunnelling, as well as the additional resource of interaction and entanglement between multiple quantum walkers [5–8]. For these reasons, it is also expected that quantum walks will be employed in applications where it would be impractical or virtually impossible to employ classical random walks due to their relative inefficiency. Indeed, quantum walks have already given rise to efficient algorithms that have applications in graph theory, quantum search, network analysis, and other areas [5–7, 9–13].

This growing interest in the theoretical applications of quantum walks has inspired many experimentalists to implement quantum walks in various physical systems with state-of-the-art techniques, including nuclear magnetic resonance [14], quantum optics [15, 16], ion traps [17, 18], neutral atom traps [19], linear optics [20–22], and integrated photonics [23, 24]. Manouchehri and Wang [25] provided a comprehensive review on the numerous proposals for a physical realisation of quantum walks, as well as actual experiments, underpinned by a wide range of quantum, classical and hybrid technologies.

In any experimental implementation of a quantum walk, error may be introduced due to the failure of any of the experiment’s components, and this error can propagate throughout the experimental system. Therefore, it is vitally important for quantum walks to be implemented fault-tolerantly; that is, in a way that ensures that error due to the failure of a single component can be corrected and will not significantly affect the result of the experiment. However, there has so far been no major attempt to achieve this fault-tolerant implementation of quantum walks. Additionally, most experimental implementations of quantum walks have so far been concerned with quantum walks on a 1D line or on simple symmetric graphs. This paper presents a method for fault-tolerantly implementing quantum walks in discrete time on any arbitrarily complex graphs.

The paper is structured as follows. A very brief introduction to discrete-time quantum

walks and the quantum walk operator U is provided in Section II A. Section II B gives a description of how the quantum walk operator U can be decomposed using the technique of cosine sine decomposition. An overview of the Steane code and a related universal set of fault-tolerant matrix operations, which are essential to the fault-tolerant implementation of quantum walks, is given in Section II C. Section II D describes how to implement a fault-tolerant quantum walk corresponding to an arbitrary U . In Section III, we present our results, including an example of a quantum walk on the 8-star graph. Finally, Section IV concludes, and provides suggestions for further research in this area.

II. THEORY

A. Discrete-time quantum walks

Consider an undirected graph $G(V, E)$, characterized by a set of vertices $V = \{v_1, v_2, v_3, \dots\}$ and edges $E = \{(v_i, v_j), (v_k, v_l), \dots\}$ consisting of connected vertex pairs. The state of a discrete-time quantum walker on such a graph is given by

$$|\psi\rangle = \sum_{(j,k) \in E} \mathcal{A}_{j,k} |j, k\rangle, \quad (1)$$

where $|j, k\rangle$ and $|k, j\rangle$ are the two states corresponding to the vertex pair (j, k) , and $\mathcal{A}_{j,k}$ are complex amplitudes. If two particular nodes j and k are not connected by an edge, then $\mathcal{A}_{j,k} = 0$ and $\mathcal{A}_{k,j} = 0$. The total number of states N required in Eq. 1 equals twice the number of edges between distinct vertices, plus the number of self loops, in the underlying graph. At each step of the quantum walk, the walker's state undergoes the transformation

$$|\psi\rangle \mapsto TC|\psi\rangle \equiv U|\psi\rangle, \quad (2)$$

where T is the conditional translation operator such that $T|j, k\rangle = |k, j\rangle$, \mathcal{C} is the coin operator which mixes the complex amplitudes of groups of states associated with each vertex, and $U = TC$ is the overall quantum walk operator. Note that T , \mathcal{C} and U are all unitary operators.

B. CS decomposition

Instead of implementing the $N \times N$ quantum walk operator U directly, which would be difficult using any physical system, it can be replaced by a series of pairwise operators using the technique of cosine sine (CS) decomposition [26–28], for example. For this scheme to work, it is required that U is an $\mathcal{N} \times \mathcal{N}$ matrix, where $\mathcal{N} = 2^M$, for some integer M . If U is not already in this form, then U undergoes the transformation

$$U \mapsto \begin{pmatrix} U & \\ & I \end{pmatrix}. \quad (3)$$

Here, I is the $(\mathcal{N} - N) \times (\mathcal{N} - N)$ identity matrix, where \mathcal{N} is the smallest power of 2 larger than N . Correspondingly, the walker's state is transformed into a state containing \mathcal{N} elements, the last $(\mathcal{N} - N)$ of which are initially 0.

After recursively applying CS decomposition $M - 1$ times to U , we find that

$$U = \prod_{i=1}^{\mathcal{N}-1} \mathcal{U}_i(d_i), \text{ where } \mathcal{U}_i(d_i) = \begin{pmatrix} u_{i,1} & & \\ & u_{i,2} & \\ & & \ddots \end{pmatrix}. \quad (4)$$

Here, each $u_{i,k}$ is a $d_i \times d_i$ block along the diagonal of \mathcal{U}_i , with $k = 1, 2, \dots, \mathcal{N}/d_i$. Each d_i can take a value in $\{2, 4, 8, \dots, \mathcal{N}/2\}$. When $d_i = 2$, $u_{i,k}$ is a 2×2 unitary matrix, and when $d_i > 2$,

$$u_{i,k} = \begin{pmatrix} \ddots & & & \\ & c_r & & \\ & & \ddots & \\ \hline & & & s_r \\ & & & & \ddots & \\ \hline \ddots & & & -s_r & & \\ & & & & c_r & \\ & & & & & \ddots \end{pmatrix}. \quad (5)$$

Each quadrant of $u_{i,k}$ is diagonal, with entries c_r and s_r corresponding to $\cos(\phi_r)$ and $\sin(\phi_r)$, where ϕ_r is some angle, and $r = 1, 2, \dots, d/2$.

If the walker's state in a quantum walk on a particular graph is given by

$$|\psi\rangle = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix}, \quad (6)$$

then $U|\psi\rangle$ can be further decomposed into a series of operations, each transforming two elements of $|\psi\rangle$, with the m th operation given by

$$|\alpha_m\rangle \equiv \begin{pmatrix} a_{p_m} \\ a_{q_m} \end{pmatrix} \mapsto \mathcal{V}_m(\phi_m) \begin{pmatrix} a_{p_m} \\ a_{q_m} \end{pmatrix}. \quad (7)$$

Here, $|\alpha_m\rangle$ consists of the two elements of $|\psi\rangle$ transformed by the m th operation, and each $\mathcal{V}_m(\phi_m)$ is either a rotation matrix about the y axis,

$$R_y(\phi_m) = \begin{pmatrix} \cos(\phi_m) & \sin(\phi_m) \\ -\sin(\phi_m) & \cos(\phi_m) \end{pmatrix}, \quad (8)$$

a rotation matrix about the z axis,

$$R_z(\phi_m) = \begin{pmatrix} \exp(i\phi_m) & 0 \\ 0 & \exp(-i\phi_m) \end{pmatrix}, \quad (9)$$

or a phase matrix,

$$\Phi(\phi_m) = \begin{pmatrix} 1 & 0 \\ 0 & \exp(i\phi_m) \end{pmatrix}. \quad (10)$$

In the case of a real quantum walk operator U , each matrix $\mathcal{V}_m(\phi_m)$ is either $R_y(\phi_m)$ or $\Phi(\pi) \equiv Z$.

C. Fault-tolerant Steane code gates

Fault-tolerant gate operations can be implemented on qubits encoded using, for example, the 7-qubit error-correcting code developed by Steane [29, 30], in a way that will allow us to implement fault-tolerant quantum walks as discussed in Section IID. Using this code, the qubit states $|0\rangle$ and $|1\rangle$ are respectively encoded as

$$\begin{aligned} |0_L\rangle = \frac{1}{\sqrt{8}}(&|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + \\ &|0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle) \end{aligned} \quad (11)$$

and

$$|1_L\rangle = \frac{1}{\sqrt{8}}(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle). \quad (12)$$

One universal set of single-qubit gates (denoted here as \mathcal{G}), which can be implemented fault-tolerantly when applied to qubits encoded using the Steane code, consists of the following elements [30, 31]:

$$H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}, \quad (13a)$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad (13b)$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (13c)$$

$$T = \begin{pmatrix} 1 & 0 \\ 0 & \exp(\frac{i\pi}{4}) \end{pmatrix}, \quad (13d)$$

$$S = T^2 = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad (13e)$$

$$s = S^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}. \quad (13f)$$

Although the subset $\{H, T\}$ of \mathcal{G} is also a universal set [30], using this subset is inefficient compared to the full set, as implementing the T gate is much more complicated than implementing the other gates in \mathcal{G} [32]. Note that the gate T^\dagger is not included in \mathcal{G} , as using this gate in gate sequences containing 2 or more gates is equally as efficient as using the gate sequences sT or Ts , which are equivalent to T^\dagger .

The fault-tolerant H , X and Z gates are implemented by applying regular H , X or Z gates respectively to each of the seven qubits corresponding to a single encoded qubit. The implementations of the fault-tolerant S gate is similar, but the regular gate applied to each qubit in this case is the conjugate transposes of the fault-tolerant gate, s . It follows that the fault-tolerant s gate can be implemented in the same way, applying the regular S gate to each qubit.

The fault-tolerant T gate requires an input of two encoded qubits, i.e. two blocks of seven qubits [30]. The first is prepared in the ancilla state

$$|\Theta\rangle = \frac{|0\rangle + \exp(\frac{i\pi}{4})|1\rangle}{\sqrt{2}}, \quad (14)$$

and the second is $|\alpha\rangle$, the qubit for which we desire the output $T|\alpha\rangle$. The ancilla state is prepared by first preparing the encoded $|0_L\rangle$ state, and then fault-tolerantly measuring $\exp(-\frac{i\pi}{4})SX$, using the procedure described by Nielsen and Chuang [30]. If the measurement gives a result of +1, then the ancilla state has been prepared. If the measurement result is -1, then the ancilla state preparation is completed by applying a fault-tolerant Z operation to the first qubit. Once the ancilla preparation is complete, a fault-tolerant CNOT gate is applied, with the first and second qubits as the control and target qubits respectively. This fault tolerant gate is implemented by applying a regular CNOT gate between each corresponding pair of single qubits in the two seven-qubit blocks. Next, the second qubit is measured fault-tolerantly, again using the method described by Nielsen and Chuang. If the result of the measurement is 0, then the first qubit is in the state $T|\alpha\rangle$, as required. Otherwise, applying the fault-tolerant S and X gates to the first qubit will transform the first qubit into the desired $T|\alpha\rangle$ state.

D. Fault-tolerant quantum walks

For certain highly symmetric graphs, exponentially efficient quantum circuit implementation of the associated quantum walks are attainable [33–35]. In these cases, the well-established fault-tolerant quantum computation protocols as discussed in [30, 36–38] can be readily applied. In what follows, however, we present a systematic method for a fault-tolerant implementation of discrete time quantum walks on any arbitrarily complex graph which, in general, cannot be implemented efficiently using a quantum circuit.

By the Solovay-Kitaev theorem [30, 39], finite product sequences of matrices in the universal set \mathcal{G} given in Eq. 13 can be used to approximate the matrices appearing in the decomposition of an arbitrary quantum walk operator U acting on the state of a quantum walker $|\psi\rangle$. The sequence length is in the order of $O(\log^c(1/\epsilon))$, where c is a small constant and ϵ is the desired precision. Since all matrices in the set \mathcal{G} can be implemented fault-tolerantly as discussed in the previous section, so can the decomposition of $U|\psi\rangle$ that

involves these matrices. In order to determine the fault-tolerant approximation of an operation in the decomposition of $U|\psi\rangle$, one needs to find a product sequence of matrices in \mathcal{G} that approximates the matrix $\mathcal{V}_m(\phi_m)$ corresponding to that operation. Then, the single operation is replaced by a series of operations, corresponding to the product sequence, each of which acts on the same state vector $|\alpha_m\rangle$. Note that in the case of a real quantum walk operator, only approximations of rotation matrices about the y axis, i.e. real rotation matrices, need to be found.

In order to find approximations of unitary matrices using product sequences of matrices in \mathcal{G} , Fowler [31] suggested finding a sequence, out of all sequences up to a certain length, that best approximates an individual matrix. With this approach, the matrix corresponding to a particular sequence is not checked against the desired matrix if the sequence is equivalent to a shorter sequence that has already been checked, and so this method is more efficient than checking all sequences exhaustively. However, since this method involves finding the approximation of a particular matrix, the procedure needs to be repeated every time the approximation of a new matrix is required. Therefore, a more efficient method, in terms of the required computational resources, is to search all possible sequences up to a certain length for those that sufficiently approximate any of the matrices $\mathcal{V}_m(\phi_m)$. In order to further maximise efficiency, all sequences and corresponding matrices are stored so that all operations on sequences and matrices during the search are based on stored sequences of shorter length, and their corresponding matrices. Rules are also implemented to ensure that sequences are ignored if they are equivalent to sequences of the same or shorter length that have already been checked. Once all sequences up to a certain length have been checked, each matrix $\mathcal{V}_m(\phi_m)$ can be associated with either a sequence that best approximates $\mathcal{V}_m(\phi_m)$ out of all checked sequences, or a sequence of the shortest possible length that sufficiently approximates $\mathcal{V}_m(\phi_m)$.

Since each two-level quantum state vector needs to be encoded as 7 qubits using the Steane code, and since each fault-tolerant T gate requires an input of two 7-qubit blocks, each operation of an element in \mathcal{G} on $|\alpha_m\rangle$ in the final fault-tolerant approximation of the decomposition of $U|\psi\rangle$ involves a $2^{14} \times 2^{14}$ matrix, acting on 2^{14} elements of the walker's encoded state vector. If the quantum walk involving these fault-tolerant operations is implemented physically, then given that the probability of error arising due to any single component of the implementation is sufficiently small, any error introduced can be effectively corrected

[30]. In this way, the results of this fault-tolerant quantum walk can be made arbitrarily close to those obtained by implementing the original quantum walk with no error involved.

III. RESULTS

Product sequences of matrices in \mathcal{G} of length up to 37 were checked, using the more efficient method described in Section IID, to find sequences that approximate real rotation matrices. A matrix corresponding to a sequence was accepted as an approximate real rotation matrix of angle ϕ if it was in the form of

$$\begin{pmatrix} \cos(\phi) + ai & \sin(\phi \pm \varepsilon) + bi \\ -\sin(\phi \pm \varepsilon) + ci & \cos(\phi) + di \end{pmatrix}, \quad (15)$$

with the condition $|a|, |b|, |c|, |d| < 0.1$. Here, ε is an error that was accepted as small enough if the aforementioned condition was met. Sequences corresponding to 1213 separate approximate angles between 0° and 180° inclusive were found, with an average difference of 0.149° between successive angles. The list of angles found included only 9 successive angle differences greater than 1° , the largest being a difference of 2.459° , and the smallest difference between successive angles on the list was 0.052° . Additionally, if a sequence of a certain length was found to approximate a rotation matrix of angle ϕ , then another sequence of the same length was found to approximate a rotation matrix of angle $-\phi$, to the same accuracy.

Fig. 1 is a plot of all approximate angles found, together with their smallest respective r values, where $r = \max\{|a|, |b|, |c|, |d|\}$. A file (*AngleSequenceTables.xlsx*) containing two pairs of tables of all angles found, together with a sequence of length up to 37 that approximates the rotation matrix corresponding to each angle, is located at <ftp://ftp.physics.uwa.edu.au/pub/QuantumComputation>. In one table of each pair, the sequence for each angle is chosen first by smallest r value and then by shortest length, and in the other, the sequence is chosen first by shortest length and then by smallest r value. One pair of tables corresponds to positive angles, and the other to negative angles. Negative angles are /d as the corresponding sequences cannot be obtained by simply inverting the sequences corresponding to positive angles, since the T^\dagger gate is not included in \mathcal{G} , as discussed in Section IIC. Although each table also includes an angle of 0° , the corresponding

sequence is not included in any decomposition of $U|\psi\rangle$, as the rotation matrix of angle 0° is the identity matrix.

We now consider, as an example, a quantum walk on the 8-star graph shown in Fig. 2. The quantum walk operator U corresponding to this graph, using the Grover Coin, is the real 16×16 matrix [28],

$$U = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.75 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & -0.75 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0.25 & -0.75 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0.25 & 0.25 & -0.75 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0.25 & 0.25 & 0.25 & -0.75 & 0.25 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & -0.75 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & -0.75 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 & -0.75 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (16)$$

This operator, acting on the state of a quantum walker, can be decomposed as described in Eq. (7), and then each matrix in the decomposition can be replaced by product sequences of matrices in \mathcal{G} . Table I lists the matrices $\mathcal{V}_m(\phi_m)$ in the decomposition of $U|\psi\rangle$, and the elements of $|\psi\rangle$ that they apply to. Table II describes the operations in the fault-tolerant decomposition of $U|\psi\rangle$. Specifically, each rotation matrix of angle ϕ in Table I is replaced with the product sequence of matrices in \mathcal{G} , out of all such sequences of length up to 37, that most closely approximates a rotation matrix whose angle is as close as possible to ϕ . In each of these tables, the operations corresponding to the first two columns appear in the overall sequence of operations before those corresponding to the last two columns, with operations later in the overall sequence being applied first, as in the case of a sequence of matrices acting on a column vector. A total of 763 matrices in \mathcal{G} appear in this fault-tolerant decomposition of $U|\psi\rangle$.

Applying the operations described in Table II to $|\psi\rangle$, directly using the matrices in \mathcal{G} instead of the corresponding fault-tolerant $2^{14} \times 2^{14}$ matrices, is equivalent to applying a

complex unitary matrix to $|\psi\rangle$ whose real part is

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.7529 & 0.2532 & 0.2436 & 0.2426 & 0.2461 & 0.2466 & 0.2470 & 0.2468 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2533 & -0.7374 & 0.2515 & 0.2411 & 0.2516 & 0.2520 & 0.2524 & 0.2522 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2433 & 0.2516 & -0.7243 & 0.2351 & 0.2493 & 0.2499 & 0.2503 & 0.2501 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2428 & 0.2411 & 0.2350 & -0.7195 & 0.2491 & 0.2501 & 0.2505 & 0.2503 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2467 & 0.2521 & 0.2499 & 0.2499 & -0.7291 & 0.2471 & 0.2466 & 0.2464 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2466 & 0.2520 & 0.2499 & 0.2501 & 0.2458 & -0.7417 & 0.2543 & 0.2361 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2463 & 0.2517 & 0.2495 & 0.2495 & 0.2438 & 0.2585 & -0.7442 & 0.2365 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2464 & 0.2518 & 0.2496 & 0.2495 & 0.2420 & 0.2374 & 0.2449 & -0.7247 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (17)$$

Comparing this with U in Eq. (16), it can be seen that the largest absolute error between corresponding elements is 0.0305, and that the largest relative error between corresponding elements is 6.00%. Additionally, the largest magnitude of an element of the imaginary part of the approximation of U , which is only nonzero in its upper-right quadrant, is 0.180. The distance between a $w \times w$ unitary matrix W and its approximation W_l is given by the metric [31]

$$d(W, W_l) = \sqrt{\frac{w - |\text{tr}(W^\dagger W_l)|}{w}}. \quad (18)$$

Using this metric, which can take values between 0 and 1, the distance between U and its approximation is 0.0901. This approximation is therefore very close to U , and so if a quantum walk is implemented using the operations described in Table II, the results of the walk will be approximately those obtained by implementing a quantum walk on the 8-star graph using U with no error involved, given that the fault-tolerant set \mathcal{G} works effectively.

It is also possible, instead of choosing the sequence that best approximates a rotation matrix of a certain approximate angle, to choose the shortest sequence that approximates the rotation matrix, again from all sequences of length up to 37. In this case, comparing the real part of the approximation of U and U itself, the largest absolute and relative errors between corresponding elements are 0.0671 and 10.8% respectively, and the largest magnitude of an element of the imaginary part of this approximation, which is again only nonzero in its

upper-right quadrant, is 0.317. Using the metric given in Eq. (18), the distance between U and this approximation is 0.121. This approximation is therefore less accurate than the previous one, and requires 735 fault-tolerant matrices, which is over 96% of the number of matrices required in the previous approximation. This indicates that longer sequences corresponding to better approximations should be used, at least in the case of this example.

IV. CONCLUSION

Quantum walks will likely play an important role in modelling and algorithmic applications in science and mathematics, but their physical implementations will only be practical when the associated experimental error can be effectively corrected. In this paper, we have presented a general method of implementing a fault-tolerant quantum walk in discrete time, using quantum states encoded with the Steane code, that approximates a quantum walk on an arbitrarily complex graph. If such a quantum walk is physically implemented, then error introduced during the implementation can be corrected effectively, as long as the probability of error associated with each component of the implementation is sufficiently small. The provided example of a quantum walk on the 8-star graph indicates that the proposed scheme can be implemented successfully. In other words, the results obtained by implementing the fault-tolerant quantum walk can be made arbitrarily close, by choosing long enough sequences of matrices in \mathcal{G} , to those obtained in the ideal case of implementing the original quantum walk with zero experimental error.

Tables of the sequences of matrices in \mathcal{G} that approximate real rotation matrices, essential to the fault-tolerant implementation of quantum walks involving both real and complex quantum walk operators, have been constructed for sequences of length up to 37. Further research in this area will focus on constructing similar tables of sequences that approximate rotation matrices $R_z(\phi_m)$ and phase matrices $\Phi(\phi_m)$, which will be required for the implementation of fault-tolerant quantum walks involving complex quantum walk operators. Additionally, the developed tables can be extended to include sequences of longer lengths, in order to improve the accuracy to which arbitrary quantum walks can be fault-tolerantly approximated.

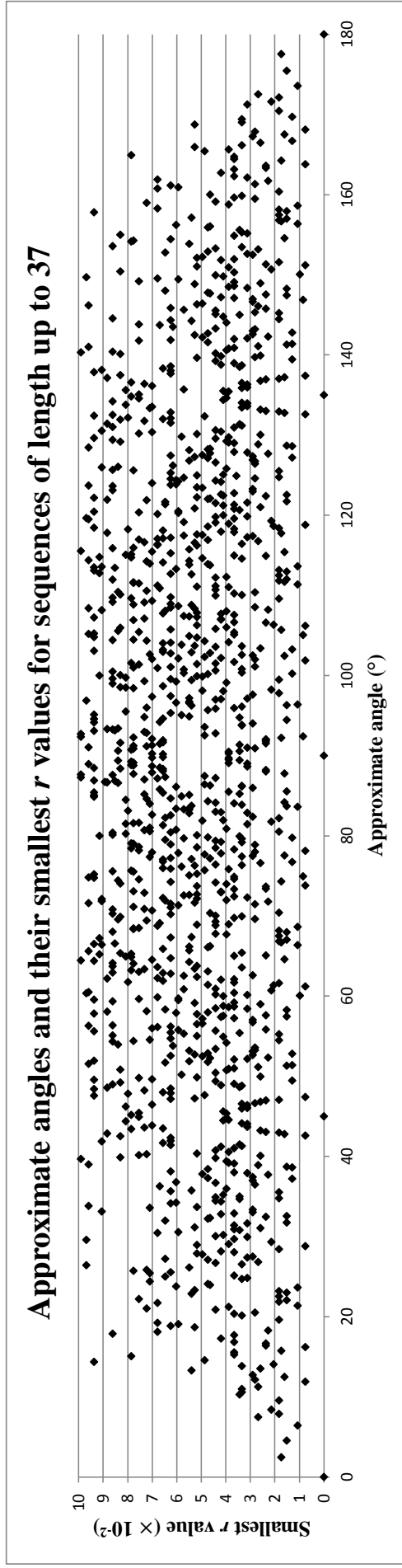


FIG. 1. All approximate angles found, and their corresponding smallest r values, for sequences of length up to 37.

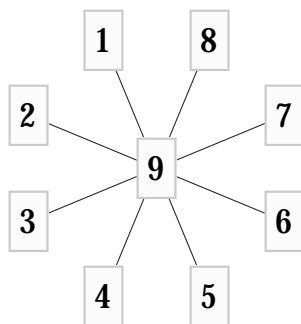


FIG. 2. The 8-star graph.

TABLE I. The matrices, and the corresponding elements in the quantum walker's state, in the decomposition of $U|\psi\rangle$. This table was generated using a modified version of the *Qcompiler* package [28].

Matrix	p, q	Matrix	p, q
$R_y(-90.000^\circ)$	1, 9	$R_y(-50.057^\circ)$	11, 12
$R_y(-90.000^\circ)$	2, 10	$R_y(-64.471^\circ)$	13, 14
$R_y(-90.000^\circ)$	3, 11	$R_y(45.000^\circ)$	15, 16
$R_y(-90.000^\circ)$	4, 12	$R_y(-90.000^\circ)$	12, 16
$R_y(-90.000^\circ)$	5, 13	$R_y(3.126^\circ)$	9, 10
$R_y(-90.000^\circ)$	6, 14	$R_y(50.057^\circ)$	11, 12
$R_y(-90.000^\circ)$	7, 15	$R_y(64.471^\circ)$	13, 14
$R_y(-90.000^\circ)$	8, 16	$R_y(45.000^\circ)$	15, 16
$R_y(18.933^\circ)$	9, 10	$R_y(-23.908^\circ)$	9, 11
$R_y(-52.595^\circ)$	11, 12	$R_y(81.630^\circ)$	10, 12
$R_y(22.500^\circ)$	13, 14	$R_y(-22.500^\circ)$	13, 15
$R_y(-22.500^\circ)$	15, 16	$R_y(67.500^\circ)$	14, 16
$R_y(23.908^\circ)$	9, 11	$R_y(-18.933^\circ)$	9, 10
$R_y(-81.630^\circ)$	10, 12	$R_y(52.595^\circ)$	11, 12
$R_y(22.500^\circ)$	13, 15	$R_y(-22.500^\circ)$	13, 14
$R_y(67.500^\circ)$	14, 16	$R_y(-22.500^\circ)$	15, 16
$R_y(-3.126^\circ)$	9, 10	Z	15, 16

TABLE II. The matrix product sequences, and the elements in the quantum walker's state, in the approximation of the decomposition of $U|\psi\rangle$. This table was generated using a modified version of the *Qcompiler* package [28].

Sequence	p, q	Sequence	p, q
XZ	1, 9	$THTHHT_sHTHTHT_sHTHTHT_sHTHTHT_sHTHTHT_sXTHTHT$	11, 12
XZ	2, 10	$sHTHTHTHTHTHT_sHTHTHT_sHTHTHTHT_sHT$	13, 14
XZ	3, 11	ZH	15, 16
XZ	4, 12	XZ	12, 16
XZ	5, 13	$ZTHTHTHZTHHTHTHTHTHT_sHTHTHTHTHTHT_sX$	9, 10
XZ	6, 14	$TH_sXTHTHTHTHTHT_sHTHTHT_sHTHTHT_sHTHTHT$	11, 12
XZ	7, 15	$THT_sHTHTHTHT_sHTHTHTHTHTHT_s$	13, 14
XZ	8, 16	ZH	15, 16
$THSTHTHTSTHTHTHTHT_sXZHTHTHTSTHTHT$	9, 10	$THTHTHTHTHTHTHTHTHTHT_sHTHTHTHTHTHTHTHTHT$	9, 11
$sHZTHHTHTHT_sHTHTHTHTHTHTSTHTHT$	11, 12	$THT_sHTHTHTHTHTHTHT_sHTHTHT_sHTHTHTHT_sHTH$	10, 12
$sH_sXTHTHTHT_sHTHTHTHTHTSTHTHTHTHT$	13, 14	$sHT_sHTHTHT_sHTHTHTSTHTHTHTSTHTHTHTHTX$	13, 15
$sHT_sHTHTHT_sHTHTHTSTHTHTHTSTHTHTHTX$	15, 16	$THTHTHZTHHTHTSTHTHTHT_sHT_sHTHTHTHT_s$	14, 16
$HTHTHTHTHTHT_sHSTHTHTHTZTHHTHTHTHT$	9, 11	$THTXZHTHTHTSTHTHTHTHTHTHTSTHTHTHTSTHT$	9, 10
$HTHT_sHTHTHTHT_sHTHTHT_sHTHTHTHTHTHT_sHT$	10, 12	$THTHTSTHTHTHTHTHT_sHTHTHTHTHTHT_s$	11, 12
$sH_sXTHTHTHT_sHTHTHTHTHTSTHTHTHTHT$	13, 15	$sHT_sHTHTHT_sHTHTHTSTHTHTHTSTHTHTHTHTX$	13, 14
$THTHTHTHTHTHT_sHT_sHTHTHTHT_s$	14, 16	$sHT_sHTHTHT_sHTHTHTSTHTHTHTSTHTHTHTHTX$	15, 16
$THTHTHZTHHTHTHTHTHT_sHTHTHTHTHTHT_sXZ$	9, 10	Z	15, 16

-
- [1] R. B. Sessions, M. Oram, M. D. Szczelkun, and S. E. Halford, *J. Mol. Biol.* **270**, 413 (1997).
 - [2] Z. T. Trautt, M. Upmanyu, and A. Karma, *Science* **314**, 632 (2006).
 - [3] B. A. Berg, *Nature (London)* **361**, 708 (1993).
 - [4] S. Hoshino and K. Ichida, *Numer. Math.* **18**, 61 (1971).
 - [5] J. Kempe, *Contemp. Phys.* **44**, 307 (2003).
 - [6] A. M. Childs, *Phys. Rev. Lett.* **102**, 180501 (2009).
 - [7] S. D. Berry and J. B. Wang, *Phys. Rev. A* **83**, 042317 (2011).
 - [8] Z. J. Li, J. A. Izaac, and J. B. Wang, *Phys. Rev. A* **87**, 012314 (2013).
 - [9] N. Shenvi, J. Kempe, and K. B. Whaley, *Phys. Rev. A* **67**, 052307 (2003).
 - [10] B. L. Douglas and J. B. Wang, *J. Phys. A* **41**, 075303 (2008).
 - [11] D. Reitzner, M. Hillery, E. Feldman, and V. Buzek, *Phys. Rev. A* **79**, 012323 (2009).
 - [12] S. D. Berry and J. B. Wang, *Phys. Rev. A* **82**, 042333 (2010).
 - [13] J. Smith and M. Mosca, in *Handbook of Natural Computing* (Springer, 2012) pp. 1451–1492.
 - [14] J. Du, H. Li, X. Xu, M. Shi, J. Wu, X. Zhou, and R. Han, *Phys. Rev. A* **67**, 042316 (2003).
 - [15] P. Zhang, X.-F. Ren, X.-B. Zou, B.-H. Liu, Y.-F. Huang, and G.-C. Guo, *Phys. Rev. A* **75**, 052310 (2007).
 - [16] P. Zhang, B.-H. Liu, R.-F. Liu, H.-R. Li, F.-L. Li, and G.-C. Guo, *Phys. Rev. A* **81**, 052322 (2010).
 - [17] H. Schmitz, R. Matjeschk, C. Schneider, J. Glueckert, M. Enderlein, T. Huber, and T. Schaetz, *Phys. Rev. Lett.* **103**, 090504 (2009).
 - [18] F. Zähringer, G. Kirchmair, R. Gerritsma, E. Solano, R. Blatt, and C. F. Roos, *Phys. Rev. Lett.* **104**, 100503 (2010).
 - [19] M. Karski, L. Förster, J.-M. Choi, A. Steffen, W. Alt, D. Meschede, and A. Widera, *Science* **325**, 174 (2009).
 - [20] M. A. Broome, A. Fedrizzi, B. P. Lanyon, I. Kassal, A. Aspuru-Guzik, and A. G. White, *Phys. Rev. Lett.* **104**, 153602 (2010).
 - [21] D. Pandey, N. Satapathy, M. S. Meena, and H. Ramachandran, *Phys. Rev. A* **84**, 042322 (2011).
 - [22] A. Schreiber, A. Gábris, P. P. Rohde, K. Laiho, M. Štefanák, V. Potocek, C. Hamilton, I. Jex,

- and C. Silberhorn, *Science* **336**, 55 (2012).
- [23] A. Peruzzo, M. Lobino, J. C. F. Matthews, N. Matsuda, A. Politi, K. Poulios, X.-Q. Zhou, Y. Lahini, N. Ismail, K. Wörhoff, Y. Bromberg, Y. Silberberg, M. G. Thompson, and J. L. O'Brien, *Science* **329**, 1500 (2010).
 - [24] L. Sansoni, F. Sciarrino, G. Vallone, P. Mataloni, A. Crespi, R. Ramponi, and R. Osellame, *Phys. Rev. Lett.* **108**, 010502 (2012).
 - [25] K. Manouchehri and J. Wang, *Physical Implementation of Quantum Walks* (Springer, in press).
 - [26] B. Sutton, *Numer. Algorithms* **50**, 33 (2009).
 - [27] K. Manouchehri and J. B. Wang, *Phys. Rev. A* **80**, 060304 (2009).
 - [28] Y. G. Chen and J. B. Wang, *Comput. Phys. Commun.* **184**, 853 (2013).
 - [29] A. Steane, *Proc. R. Soc. Lond. A* **452**, 2551 (1996).
 - [30] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information : 10th Anniversary Edition* (Cambridge University Press, Cambridge, 2010).
 - [31] A. G. Fowler, *Quant. Inf. Comput.* **11**, 867 (2011).
 - [32] P. Aliferis, D. Gottesman, and J. Preskill, *Quant. Inf. Comput.* **6**, 97 (2006).
 - [33] B. L. Douglas and J. B. Wang, *Phys. Rev. A* **79**, 052335 (2009).
 - [34] T. Loke and J. B. Wang, *Comput. Phys. Commun.* **182**, 22852294 (2011).
 - [35] T. Loke and J. B. Wang, *Phys. Rev. A* **86**, 042338 (2012).
 - [36] P. Shor, in *Proc. Symp. on the Foundations of Computer Science* (IEEE Press, 1996, Online preprint quant-ph/9605011).
 - [37] A. M. Steane, *Phys. Rev. Lett.* **78**, 2252 (1997).
 - [38] J. Preskill, *Proc. R. Soc. Lond. A* **454**, 385 (1998).
 - [39] C. M. Dawson and M. A. Nielsen, *Quant. Inf. Comput.* **6**, 81 (2006).